Combining Spatial and Telemetric Features for Learning Animal Movement Models

Berk Kapicioglu (Princeton University), Robert E. Schapire (Princeton University), Martin Wikelski (Max Planck Institute for Ornithology),
Tamara Broderick (University of California, Berkeley)

Abstract

We introduce a new graphical model for tracking radio-tagged animals and learning their movement patterns. The model provides a principled way to combine radio telemetry data with an arbitrary set of user-defined, spatial features. We describe an efficient stochastic gradient algorithm for fitting model parameters to data and demonstrate its effectiveness via asymptotic analysis and synthetic experiments. We also apply our model to real datasets, and show that it outperforms the most popular radio telemetry software package used in ecology. We conclude that integration of different data sources under a single statistical framework, coupled with appropriate parameter and state estimation procedures, produces both accurate location estimates and an interpretable statistical model of animal movement.

The State Space Model (SSM)

Let $x_t \in Q \subset \mathbb{R}^2$ denote the latent animal locations at time t. Q is a discrete set.
Let $f_k : Q \times Q \to \mathbb{R}$ denote the kth feature function and $\lambda_k \in \mathbb{R}$ denote the corresponding weight.
Let $y_{t,n} \in [-\pi, \pi)$ denote the radial bearing observed by tower n at time step t.
Let **boldface** denote the vector representation of corresponding parameters and random variables.

Start Model (uniform): $p(x_1) = \frac{1}{|Q|}$. Transition Model (Gibbs):

$$p(x_{t+1}|x_t;\boldsymbol{\lambda}) = \frac{\exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_{t+1})\right)}{\sum_{x_q \in Q} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_q)\right)}.$$

Observation Model (Von Mises):

$$p(\boldsymbol{y}_t|x_t;\boldsymbol{\mu},\boldsymbol{\kappa}) = \prod_{n=1}^{N} p(y_{t,n}|x_t;\mu_n,\kappa_n) = \prod_{n=1}^{N} \frac{\exp(\kappa\cos(y_{t,n} - h(x_t, z_n) - \mu_n))}{2\pi I_0(\kappa)}.$$

SSM:

$$p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\kappa}) = p(x_1)\prod_{t=1}^{T-1} p(x_{t+1}|x_t;\boldsymbol{\lambda}) \prod_{t=1}^{T} p(\boldsymbol{y}_t|x_t;\boldsymbol{\mu},\boldsymbol{\kappa}).$$

Parameter Estimation:

$$\left(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\kappa}}\right) = \underset{\boldsymbol{\lambda}\in\mathbb{R}^K, \boldsymbol{\mu}\in\mathbb{R}^N, \boldsymbol{\kappa}\geq 0}{\arg\max} \log p(\mathbf{y};\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\kappa}).$$

Location Estimation:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in Q^T}{\arg\max}\, p\left(\mathbf{x}|\mathbf{y};\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\kappa}}\right).$$

Stochastic Gradient versus Expectation-Maximization

We introduce an efficient stochastic gradient (SG) algorithm to handle the computational challenges of incorporating a ***very very large*** latent state space. Both SG and EM are iterative algorithms and it is hard to predict beforehand how many iterations are necessary until convergence. However, a comparison of each iteration of a *direct* implementation of both algorithms shows that SG is asymptotically superior to EM.

Let T be the number of time steps, K be the number of features, Q be the total number of hidden states, and V be the unique hidden states visited by the MCMC sample – note that MCMC is only used by SG.

EM: $\Theta(Q^2 T)$ **SG:** $\Theta(K(QV + T))$

Under the assumptions K << T and K << Q, which are valid for our problem, each SG iteration is cheaper than an EM iteration.

Synthetic Data Experiments

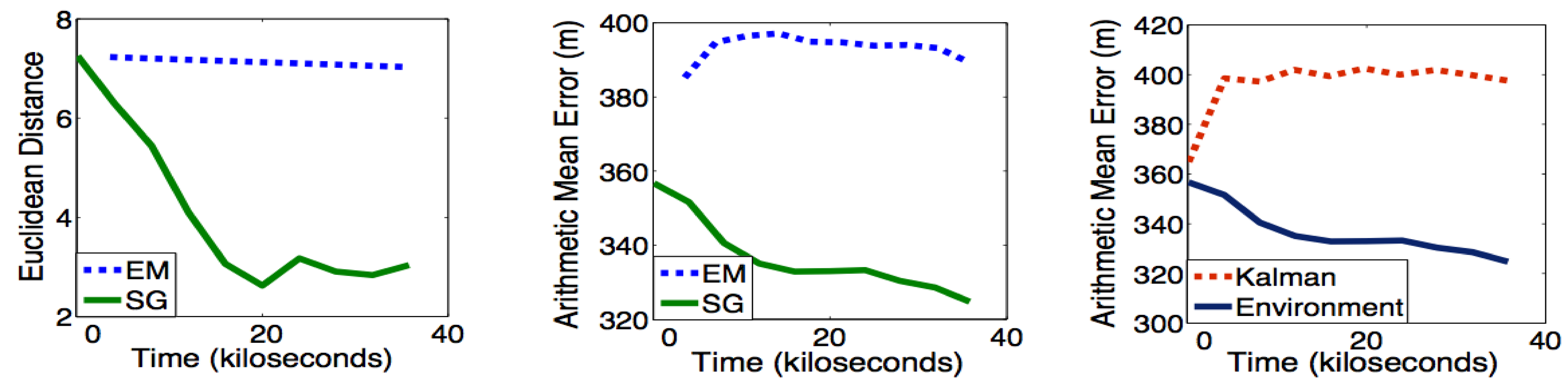

Figure 1: ***Left and middle***: *In these plots, we compare the performance of the EM and the SG algorithms. During 10 hours, EM had an average of 10 iterations, whereas SG had an average of 500 iterations. The left plot reports the Euclidean distance between the learned weights and the true weights, and the middle plot reports the average mean error of the location estimates. SG outperforms EM in both cases.* ***Right***: *In this plot, we compare the performance between using a richer feature-based (Environment) model versus using a simpler random walk (Kalman) model. The results imply that, if the animal indeed moves around based on environmental features, a model that incorporates such features does yield better location estimates than a simpler random walk model.*

Real Data Experiments (Sloth Dataset)

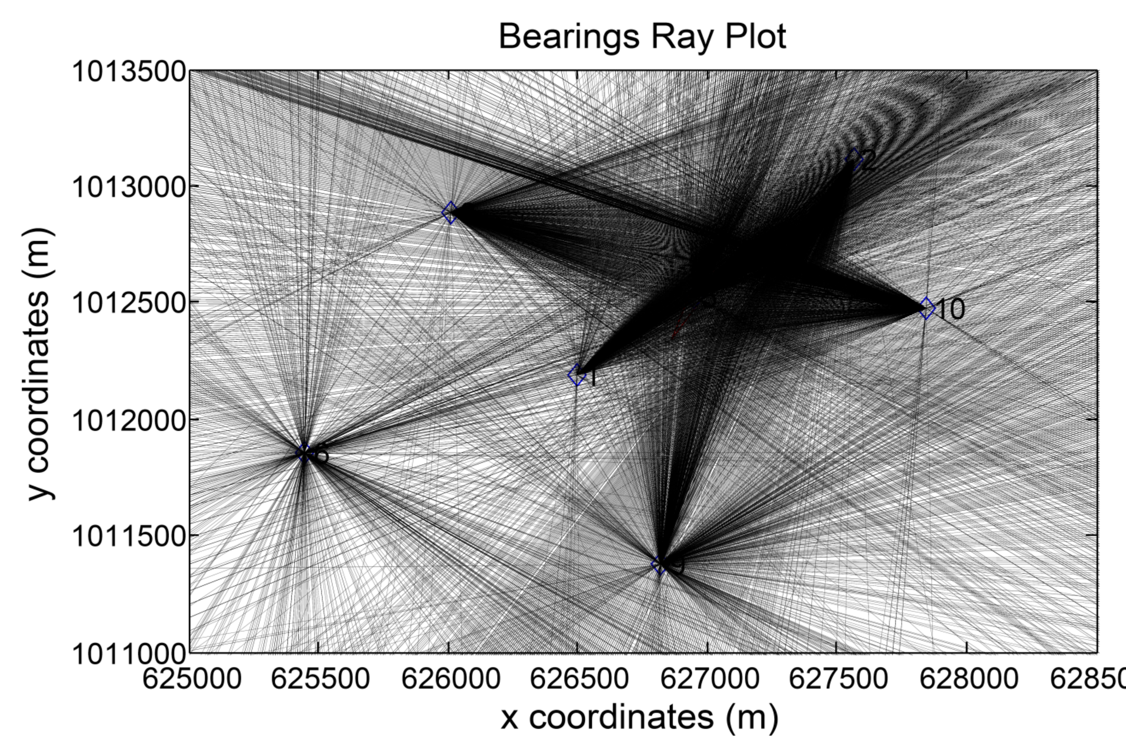

Figure 2: *This plot demonstrates the amount of noise present in our datasets. The blue diamonds indicate the tower locations. For each bearing that has been observed over a 10 day period, there is a ray coming out of the corresponding observing tower. Note that many of the rays point to regions that are far from the true location of the animal.*

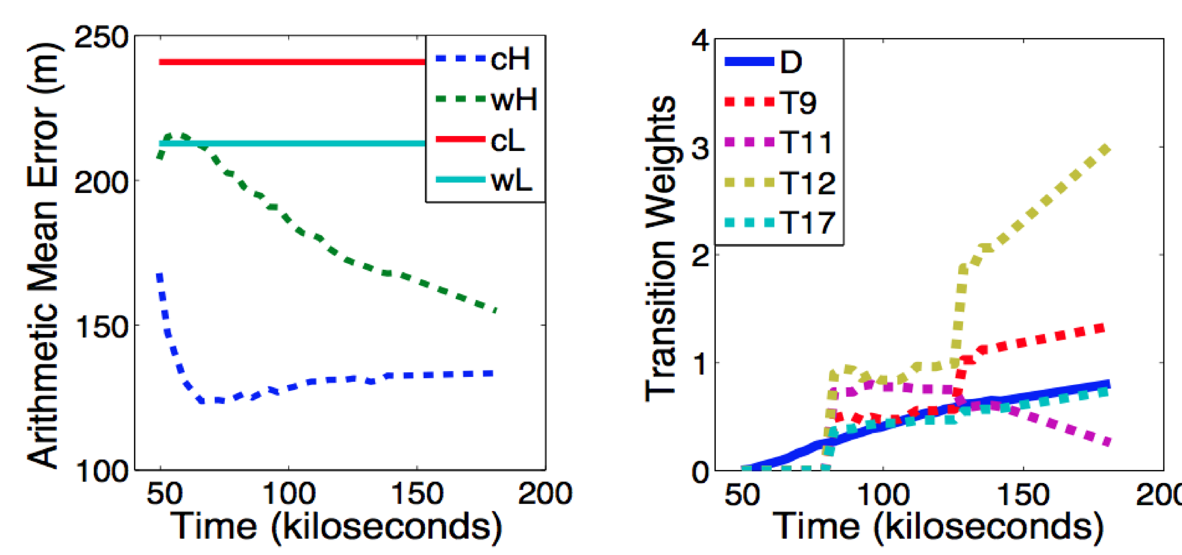

Figure 3: ***Left:*** *This plot compares the location estimation performance of the two algorithms. cH and wH represent the performance of the SSM; cL and wL represent the performance of LOAS. The initial letters "c" and "w" refer to the Chispa and Wendi datasets, respectively.* ***Right:*** *This plot displays the evolution of the transition weights for the Wendi dataset. We only plotted the weights that exceeded the 0.5 threshold. "D" and "T" denote the distance-based and tree-based features, respectively.*

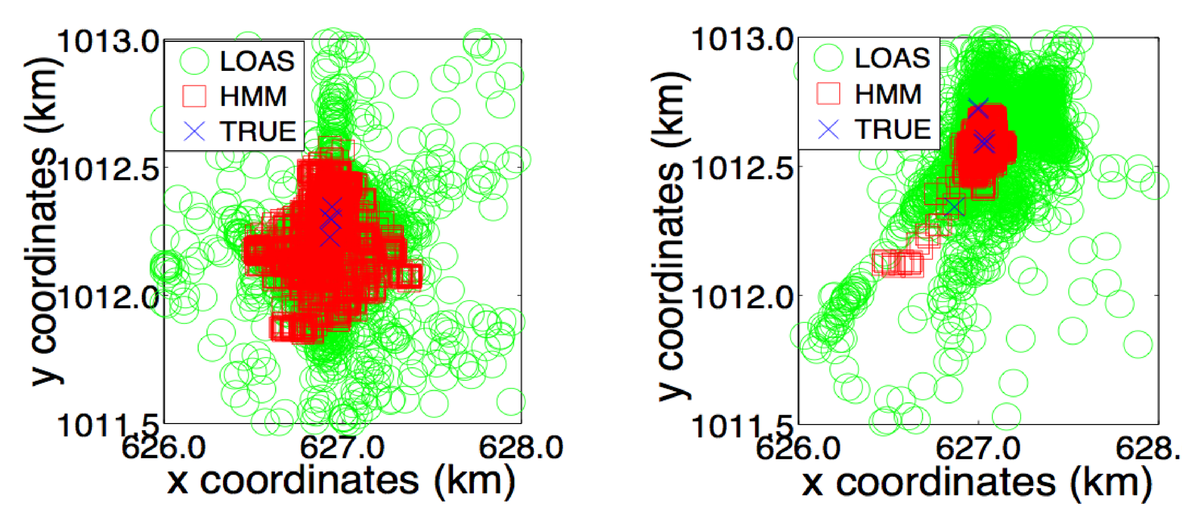

Figure 4: *These plots display the true locations, SSM estimates, and LOAS estimates. SSM estimates are based on the last stochastic gradient iteration. Left plot displays the results for the Wendi dataset and the right plot displays the results for the Chispa dataset.*